Product Features

The Good, the Bad, & the Ugly

uservoice

Contents

03
Introduction

05
The Beginning

16
The Good

34
The Bad

42
The Ugly

53
Conclusion

Introduction

Everybody loves a good feature. Engineers love building them, sales loves showing them off, marketing loves having a new selling point to work with, product managers love improving their product's capabilities, and most importantly, when features are done right, customers love using them.

Any old product team can manage the development and implementation of a feature, but truly exceptional product managers know that executing great care throughout every feature's lifecycle is the key to owning a feature set that rocks. Many product managers make the mistake of believing that the only parts of the development lifeycle that matter are prioritizing, building, and launching, but great product managers know that that is just the beginning of the process.

What to Expect

This eBook provides **INTERMEDIATE-LEVEL** information for product managers who are seeking knowledge and advice on improving existing products by strategically adding new features and functionalities.

In this eBook, we'll address:

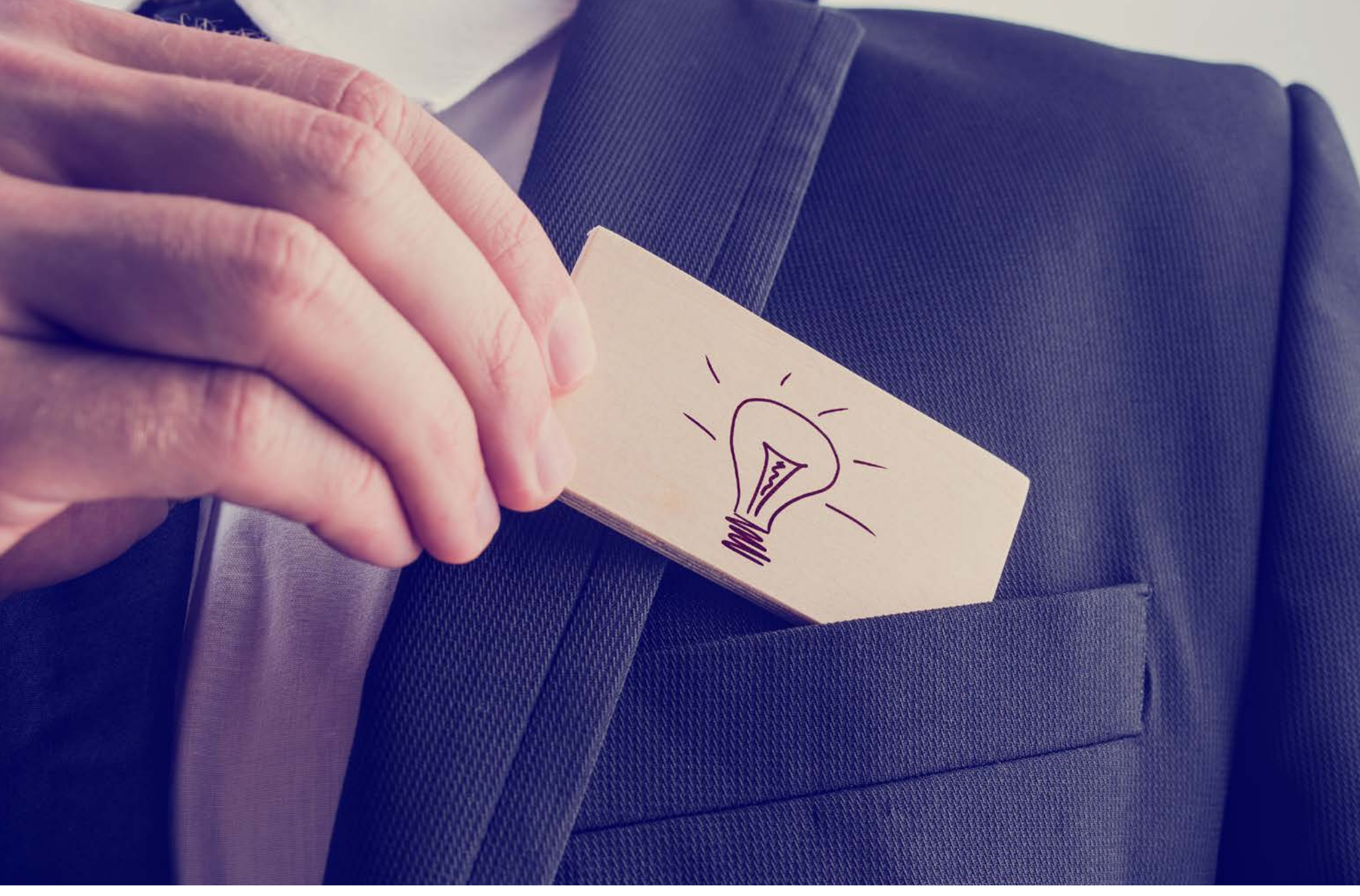
- How to ensure you're building the right features.
- The true costs of implementing and maintaining a feature.
- When and how to say "no" to a feature request.
- How to build and launch successful features.
- How to rescue features that flop
- What to do with features that simply aren't making the cut.

01

The Beginning

Where Do (Good)

Features Come From?



The Beginning

Where Do (Good) Features Come From?

Customers, executives, sales, developers, support, stakeholders–each of these groups will likely suggest you implement a feature at some point or another. If you listened to and acted on every suggestion you received, your product would get very feature heavy very quickly.

TWEET THIS

Don’t try to make your product into a Swiss Army Knife

Unless your product is a Swiss Army Knife, it was likely built with one core function in mind and should focus first and foremost on doing that one thing well; features are extra, the cherry on top of an already great sundae. For this reason, it’s critical that you stay focused on your product vision, your product’s purpose–focus on the basic need that your product seeks to serve and the long-term goal of your product.

Before you give a feature request the thumbs up, you must carefully consider whether it’s a commitment worth making. At the most basic level, ask yourself whether the feature will help you achieve your product vision or other critical business goals (including those that relate to customer happiness), or whether it’s just “cool.”

PRO TIP: Choose Features for More than Just the “Cool” Factor.

The source and weight of a request is also important to note: your

dev team, sales team, and stakeholders aren't your customers, so while their ideas may be on par with product and business goals and they most likely have some understanding of your customers, you'll want to validate these ideas with actual customers.

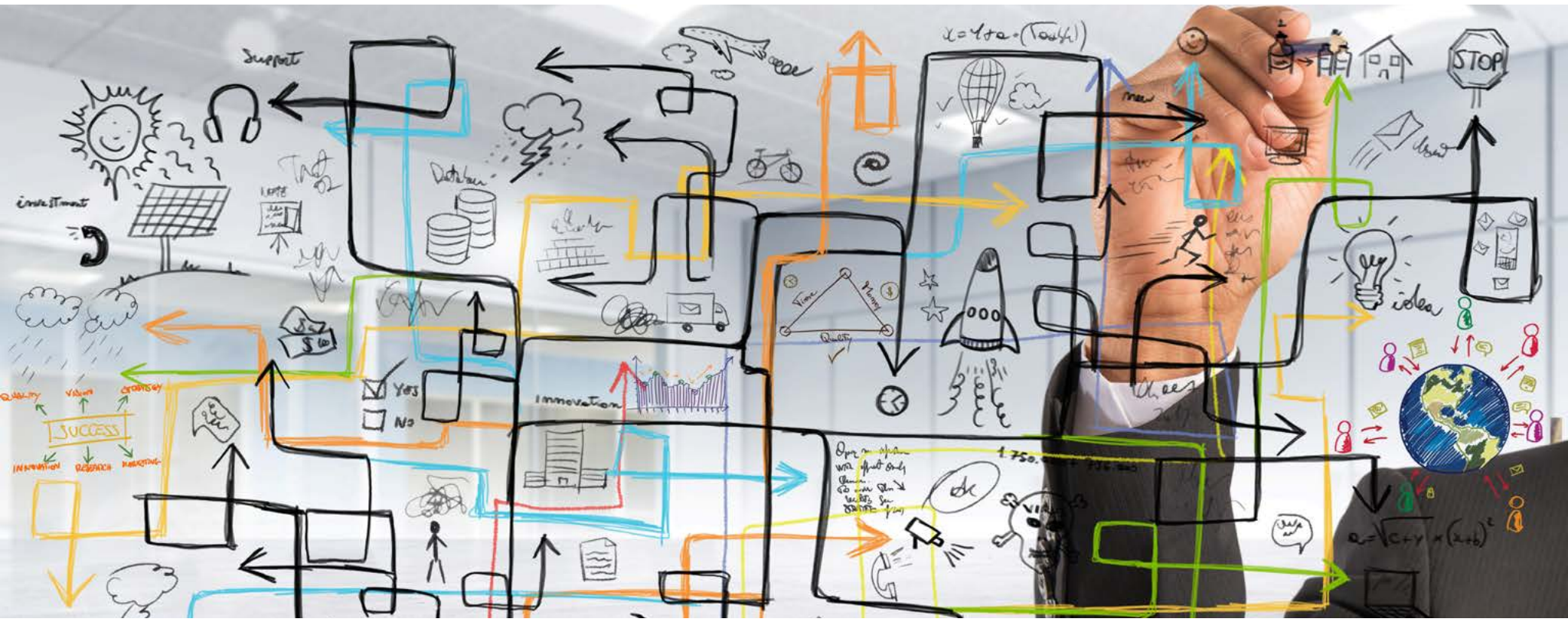
Just as a feature suggestion from the sales team or an executive shouldn't be implemented merely due to its source, a request from a single user shouldn't be considered validated either. Don't feel obligated to build something just because one customer requests it. You should be looking at the entire market, not just listening to feedback from the vocal minority.

The Cost Commitment

Features Aren't Free to Build (or Keep)

In addition to determining whether there's demand for a new functionality, you may also want to consider the true cost of implementing and maintaining a feature as well as where it fits in with other product needs before committing.

It's easy to underestimate the true cost of adding a new feature to your product, and even easier to forget that development and implementation costs are just the beginning. Once you've built and shipped something, you also must budget to maintain it; bug fixes, updates (and QA testing), documentation, and support staff hours are often left out of initial estimates of the ongoing costs of new features.



Furthermore, there's also complexity costs to factor in. With every extra tool or functionality we add to our products, we also add a level of complexity; a UI learning curve, another button on the dashboard, an extra page or two in the manual, all of these things must not disrupt the key function of our products.

Prioritization & Balance

Factoring in Time and Resource Limitations

In addition to considering the costs involved with bringing a new feature into the world, it's imperative to ensure your priorities are in the right place when deciding to add (or not to add) a new feature. If your product is drowning in technical debt that will demand several

cycles of infrastructure updates from your dev team to fix, now is not the right time to build something fancy and new. Much of this is a matter of maintaining a healthy product roadmap that balances both product innovation and product upkeep.

PRO TIP: Feature Requests from Customers Come First (But They Aren't Always Right).

Customers don't always know what's best for them, but their feedback can provide insight into the underlying problem (or, often, problems) they're experiencing and indicate unmet needs. The key word here is "needs." It's critical to understand that what your customers want is not always what they need, and where there's no underlying problem, there's no underlying need. Focus first on customer needs, and second on wants.

TWEET THIS

Customer wants aren't synonymous with customer needs

Don’t Load Up on Junk Features

Just Say “No” to Feature Bloat

Focusing on needs also means protecting your customers (and product) from too much of a good thing. Pack too many features into a product and you risk subjecting users to “feature fatigue,” a term coined in a 2009 study published in the Journal of Marketing Research that describes the feelings of frustration, defeat, and buyer’s remorse experienced by consumers who purchase feature loaded products.

The study looked at how features influence both buyer behavior and customer satisfaction and found that consumers often bite off more features than they can chew when making purchases. It concluded

"Too many features can encourage initial purchase but damage satisfaction and reduce repurchase probabilities, leading to lower customer lifetime values."

- THOMPSON, HAMILTON, & RUST
"Feature Fatigue"

that extra bells and whistles attract interest and may entice shoppers to go for one product over the other--but those features don't always get used and buyers tend to vastly overestimate their ability to learn to use the extra features they buy, leading to frustration and a degraded user experience.

It takes a focused, assertive Product Manager to keep a product on track. It's critical to cut through the often-deafening noise of feature requests and asks to avoid getting swept up into the chaos of "add this!" "build that!" and "more!" Plan and execute every feature with great care; have a clear vision of long term goals in mind and be selective about what feedback you act on.

Advocating for customers and focusing on the big picture means saying "no," often. A good product manager is not only comfortable with the word "no," but is also prepared to follow up with "why" in an audience-appropriate manner. Internally, this may mean bringing out some data to back up your reasoning.

PRO TIP: Saying "No" to Customers is Better than Ignoring them

Ignoring customer feedback is worse than saying "no." Don't leave customers hanging, acknowledge and thank them for their sugges-tions, and be honest if you're not going to build something they've requested. Ideally, you'll not only tell customers you're not going to act on a request, but also why you're not going to build the specific thing they've requested.

TWEET THIS

Ignoring customer feedback is worse than saying "no."

"Being assertive during the design process brings value in the long run. It keeps your product clean, easy-to-use and worth recommending. The problem is, people are more comfortable with making short-term decisions. Junk food and cigarettes would not be so popular otherwise. Long-term vision is something that a good product manager must always keep in his mind."

- BARTOSZ OLCHÓWKA
LiveChat, "Beware of Feature Overload"

02

The Good

Building and Launching
Features that Succeed

The Good

Building and Launching Features that Succeed

Even the most brilliant feature ideas are not immune to failure, and the decisions you and your team make between idea and launch can have a major impact on a feature's outcome. During this critical period in your feature's lifecycle, there are several things you can (and should) take care of to improve your feature's likelihood of success.

Before You Build

Don't Forget to Mind Your Metrics!

Metrics often come into play while teams decide to build (or not to build) a feature, but once your team gives a new feature the green light and begins working to implement it, data becomes increasingly more important as it's an indicator of both the health of a feature (is it being used? does it work?) and its success (is it helping you get closer to X business goal?) For this reason it's wise to identify a feature's vital signs; the metrics that will help you assess its health and success over time, before you begin building, and start tracking them as early as you can.

"The moment you [start] specing it out, you should know exactly what metrics you want to measure. As a baseline standard, you want: Engagement, Time Spent, and probably Conversions if you're an e-com site, but then for the exact product, know what you're going to measure and then make sure that you're aware of it and that you are also willing to completely and utterly disregard it if it isn't something that people love and will keep coming back to your site for."

- DWIGHT CROW
Facebook, "Be Data Informed"

Identifying the Metrics that Matter

Your feature's key metrics will depend on various factors including your product vision, business goals, and the nature of the feature itself, but it's generally a good idea to consider covering at least two categories:

METRICS THAT MEASURE A FEATURE'S GREATER IMPACT

These are metrics that look at the "big picture," and help you quantify how a feature is moving the needle on business objectives, or helping your product or organization progress toward its overarching goal. These metrics are the ones that matter most to stakeholders and your organization as a whole.

METRICS THAT MEASURE A FEATURE'S "VITAL SIGNS"

This category most often contains usage and usability metrics; figures that relate to short-term goals or unique goals of each individual feature. These metrics may come in handy in helping you determine whether a feature gets to stay or go when you do some feature "clean up" down the line. They're also useful tools for identifying problems, UI pain points, and finding ways to iterate and improve it over time.

PRO TIP: Try HEART for Feature Vitals

If you're struggling to pick the proper feature-level metrics, exploring Google's HEART framework for UI metrics may be a good starting point:

- **H** Happiness Metrics (like user satisfaction scores)
- **E** Engagement Metrics (like average visits or uses per user)
- **A** Adoption Metrics (like new users)
- **R** Retention Metrics (like churn)
- **T** Task Success Metrics (like form error rates)

While You Build

Collect Feedback, Iterate, Repeat

You can build and launch a feature that solves the most painful of customer problems but if it does not solve the problem the right way, have you really solved the problem?

In product management, it's not enough to simply identify the solution to a problem--you must also identify (and build) the right solution to that problem, which is why it's wise to bring real users into your development process early on. Engaging customers as you build and iterating based on their feedback is one of the best measures you can take to ensure a feature's success.

TWEET THIS

Engage users as you build and use their feedback to guide changes

Whether you're just beginning to hammer out the details of a new feature, or you're finalizing the copy on the UI, there's plenty of opportunities for you to collect feedback--seize them, because incorporating insight from customer feedback, interviews, surveys, and usability studies into your process can make the difference between a feature (and product) that soars and one that lands in the market with a dull thud.

Before You Launch

Pre-Feature Launch To-Dos

Repeat after me, "Just because it's been built doesn't mean it's ready to launch." Even if you have management and stakeholders breathing down your neck and pressuring you to launch, it's in your best interest to take a little extra time and check all the boxes before shipping.

When you are planning a release, put some wiggle room between your feature's code complete and ready-to-launch dates. Not only will this help ensure you don't launch anything prematurely, but it

TWEET THIS

"Built" does not equal "Ready to Launch"

also will minimize internal stress between product, support, and marketing teams before a big launch.

Every launch, big or small, requires coordination across several teams, and there's a few very important things you should take care of before launching any new feature or functionality including:

1. Training Your Teams
2. Producing Marketing Collateral
3. QA & Testing
4. Producing Documentation
5. Getting The Right Tracking Systems in Place

Training Your Teams

Everybody within your organization who interacts with your product and/or customers should be trained before launch. Training isn't just a support or sales issue (although customer-facing staff should definitely know how to answer questions about new features), it's also about evangelism.

Every sales call, demonstration, and conversation about your product holds the potential for your new feature to be mentioned or shown off. But if your staff doesn't know enough about it or feel comfortable talking about it, they will likely shy away from bringing it up.

MARKETING

No matter how amazing your new feature is, if marketing doesn't know how it works then they won't be able to properly articulate the capabilities to create that spark in the market. You won't get press or analyst coverage. Forget to train them and you could miss your chance to create a splash.

SALES

Don't forget to train sales. There's two types of salespeople; type A rushes ahead to sell every new thing they hear about, and whatever capability the customer asks about, the answer is always "yes." How are they supposed to know it won't do that if you haven't taken the time to teach them what it does do?

Type B salespeople take the exact opposite approach. If they don't know what it does, they won't mention the feature or call it to the attention of your customers.

SUPPORT

Skip training the support team and you'll definitely regret it. When customers can't get their questions answered, you have unhappy customers. Unhappy customers beget a skittish sales team, and before you know it, another urban legend is born, "Rumor has it the feature doesn't work"—so the sales force never mentions it again.

PARTNERS

Finally, depending on your business model, you may also have a network of sales agents or outside distributors selling your product, don't forget about your partners! There's nothing more painful than shadowing a rep on a sales call and realizing they are still following the talking points and feature set they learned about three years ago.

PRO TIP: Pre Launch Training Doesn't Have to Be a Huge Production

In some cases, an internal email describing the new feature or linking to a screencast of it in action will be sufficient training on a new feature, but in many cases a hands-on approach will be far more effective. Depending on the size of your organization, you can try internal webinars or lunch-and-learn sessions, even very basic training is better than none.

Producing Marketing Collateral

The marketing team can be your greatest allies, but only if you let them. It's wise to begin consulting with marketing well before you prepare to ship something. They tend to have a good grasp on what your customers want, and just as importantly they tend to know what the market is humming about. You'll want their help with placement--they'll know the right terminology; the right buzzwords, and they'll know what the competition is saying.

Working with marketing early on gives them time to develop key messages, and to incorporate the messages into sales and promotional collateral, which will also be an important part of your launch.

PRO TIP: Leverage Press to Generate Pre-Release Buzz

If people are always looking for the latest bit of info about your product, a well-timed press "leak" can generate some pre-release buzz that will have people ready to start using your new feature the moment it's launched–and in many cases it will bolster the amount of press you get when you launch as well.

Consider a preliminary release 3 weeks or so before your actual release and launch to get your audience excited, when you launch your new feature, you'll already have some users who've been eagerly awaiting it since the news first hit the wires.

This technique won't work for every product or feature, and it's never guaranteed that a press release will pick up coverage, nor does getting said coverage guarantee you'll generate buzz.

An alternative option would be to reach out to your most influential customers and give them a "sneak peek" of your new feature before it's released, it's a great way to collect early feedback and possibly get them talking to other people about what you're about to release.

QA & Final Testing

Nothing should go to market without going through a final round of testing, and you can't test a feature until it's complete. Although this may seem intuitively obvious to developers, it's a slightly foreign concept to many folks outside of R&D. Your launch process must include adequate time for final tests, no matter how much unit testing and usability testing you do along the way.

Not only should you should test that the code executes as planned, it's also a good idea to do a final round of testing with your customers; whether that means releasing a feature in beta to a small group of users, or holding another round of usability testing. The point is to ensure customers and potential users agree that the final "product" still meets their needs and solves the problem you initially set out to solve. The amount of time you spend on a feature's final round of testing will vary from release to release depending on complexity.

TWEET THIS

Don't launch anything without a final round of user testing

Producing Documentation

While the days of thick binders of product documentation are long gone for most products, you will still need some form of documentation, knowledge base articles, etc. for your customers. Depending on the complexity of the feature you're launching, you may want to incorporate documentation and other, similar content into your usability testing to make sure that you've thoroughly covered what you need to in a way that's easy for users to understand.

Getting the Right Systems in Place

Launching is not just a "push-button" event but a process that runs on a continuum, you must be prepared not only for things to go

SHARE

wrong (they will), but also to change things that aren't working. For this reason, it's helpful to have a system in place to collect and track feedback from customers before you launch. Remember those metrics we discussed earlier? Don't forget about those; have a system in place for tracking those metrics before you ship.

PRO TIP: Don't Panic About Early Usage Metrics

Speaking of metrics, you may be in for a bit of a shock when you first launch. Don't let early metrics get you down. At this stage, the feature's "introduction" as it would be called on many Product Lifecycle diagrams, expect extremely low usage metrics. Not everyone is an early adopter, and what's more is many users take their time getting used to new features.

And if those usage metrics never improve? Don't panic, you can still recover. The next chapter covers a few easy options.

03

The Bad

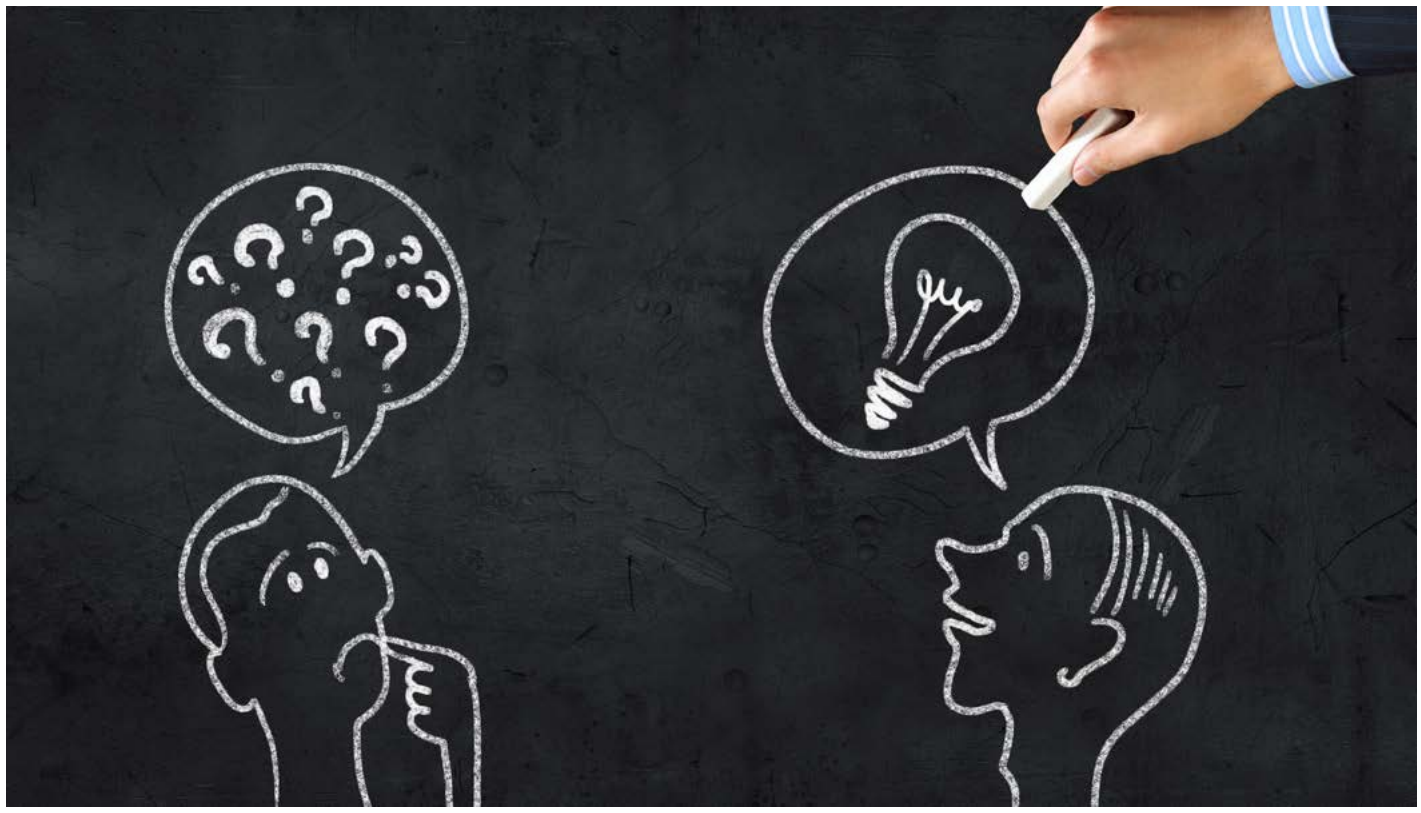
You Built it...

But No One Came, Now What?

The Bad

You Built it...But No One Came, Now What?

Unfortunately, just because you (and your data-driven rationale) believe your feature will be a hit with customers, there's no hard guarantee that people will actually use it when it's released. So what happens if even after doing all the market research, conducting the customer interviews, and checking all the boxes, you launch a feature that simply doesn't get used? There's a few ways you can remedy a feature flop at various points in a feature's lifecycle, and they all have one theme in common: customers.



1. Talk to Your Customers

Unless you're just making things up yourself (which we all know is a terrible product management strategy), at some point a customer likely declared, "It'd be really great if your product could peel bananas while updating the CSS structure," or suggested whatever feature it is that you've now built.

For starts–if you didn't talk to this person (or group of people) while you were planning, building, or testing your feature, consider trying to do that next time. The extra effort required to gather and store customers' email addresses is worth it. If you missed the opportunity to involve these users in the development process, reaching out to them shortly after launch is your second best option. Once your product actually DOES peel bananas while updating the CSS

structure, follow up with the people who initially suggested or supported you implement such a feature and show it to them. It isn't just good customer service, it's an opportunity for you to rescue a flailing (or straight up failing) feature; be prepared for the worst and expect the best.

WORST CASE SCENARIO: Users Tell You it's Terrible.

Your worst case scenario isn't actually as bad as it sounds. If your customers tell you your new feature is terrible, you won't be left wondering why it's not being used and you can continue engaging with your users while you build a solution that they actually like.

BEST CASE SCENARIO: You've Got New Promoters.

Best case scenario, you share the new feature with users and they think it's fantastic; they not only begin using it themselves but also share it with others. Customer evangelism often trumps marketing efforts.

Finally, you can leverage these conversations with your users to generate testimonials, case studies and other customer-driven marketing pieces that can be used to tout the benefits of your new feature–it's a win-win situation.

2. Let Customers Test Drive it for Free

Unless your new feature costs your company money each time it is used or every time you add a new user or seat, there's plenty of incentive to give people a free trial. A free trial helps diminish a user's fear of committing to something they won't like or use, and hey, you might even get people hooked who would have ordinarily never even given it a spin.

If the ultimate goal of this particular new feature is to deliver value and incremental revenue, you can offer a time-based free trial (30-90 days) or create a free tier with a usage cap that will get the real users to upgrade and start paying a premium.

PRO TIP: Find Out Why Customers Don't Stick Around.

Bonus: if customers are opting out after their trial period, find out why. Opt-outs and cancellation emails are the perfect opportunity for you to collect feedback and identify the source of a particular feature's trouble.

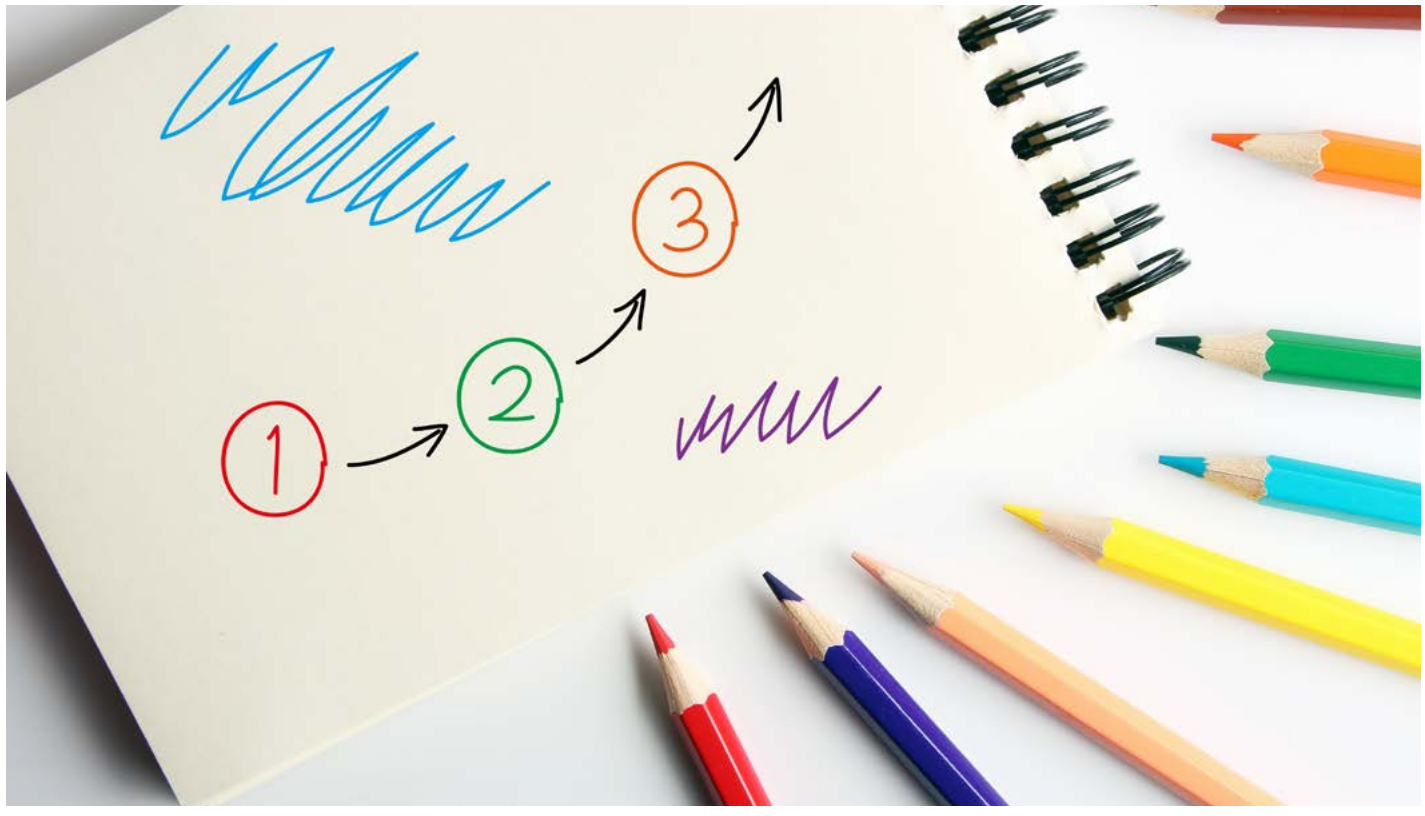
3. Show Off Real World Applications

Any new feature that has a learning curve (read: requires a user actually do something to utilize it) as opposed to those that make your product faster, brighter, lighter, more durable etc. is well served by having its attributes communicated in terms of how they can be used in the real world.

One of the easiest ways to show your customers the real world applications of your feature is to take the "eat your own dog food" approach and demonstrate how your own business is using the new functionality in a blog post or other medium your customers will see.

4. Publish Tutorials and How-Tos

Just because you have added a new feature and had your UI/UX gurus come up with the best navigation ever doesn't mean a potential user is going to feel comfortable diving in without some guidance.



TWEET THIS

Bite-sized tutorials are a great way to get users to try new features

Having your team schedule regular customer success calls to show customers the ins and outs of new features and help them start using them may be a smart move, but success calls take time and your team may not be able to reach every customer. For the rest of your users, a step-by-step tutorial is probably the most effective way to help them get started.

In some cases, feature video tutorials will make a lot more sense than blog posts or other written formats. The key to these is making sure they are as bite-sized as possible, don't try to cover everything your product has to offer in one long and comprehensive video.

PRO TIP: For Best Results, Promote and Repurpose Your Tutorials

Don't think just because you created a great tutorial people will read it or watch it – you should make sure it is available wherever it is relevant on your site and/or within your product itself, and be sure to work it into your onboarding procedure.

04

The Ugly

What to Do When Features
Threaten Product Function

The Ugly

What to Do When Features Threaten Product Function

As I mentioned in Chapter 1, features are not free; they come with complexity costs and require ongoing maintenance and support (even if they aren't necessarily being used), and sometimes despite our best judgement and our best efforts, features simply don't end up performing as well as anticipated; sometimes our products get

cluttered with features that aren't adding much value; sometimes a little feature Spring cleaning needs to happen–and that's okay, in fact it's encouraged.

Much like Massimo Vignelli proclaimed in Helvetica, "The life of a designer is a life of fight. Fight against the ugliness. Just like a doctor fights against disease," the life of a Product Manager is a life of fight as well; a fight against the ugliness that is clutter and noise.

TWEET THIS

Get rid of feature dead weight as often as you need to

"Perfection is attained not when there is nothing more to add, but when there is nothing more to remove."

- ANTOINE DE SAINT-EXUPERY
"Terre des Hommes"

Product Managers must be both fearless and objective when removing features that don't support the bottom line and swift to either remedy or kill features that drain resources or user patience.

Get rid of feature dead weight as often as you need to. It's better to kill off an unwarranted feature (or features) than to let it pull the whole product down to the grave.

There are two main categories of problem features to look out for:

1. **EXCESS FEATURES**
2. **FORGOTTEN FEATURES**

"Your users probably have very little time to sift through piles of options or user guides to figure out how to accomplish what they signed up for in the first place. If they can't figure this out quickly and easily, they'll walk away."

- JANNA BASTOW

Mind The Product, "To Kill a Feature"

Excess Features

High Complexity Costs & Low Value to Product

If new customers are taking one look at your product and leaving, or worse, unsuccessfully attempting to use it for its core function and giving up out of frustration or confusion, that may be a good indicator some feature clean up is in order.

When you add features and functionalities to your product, you also add a layer of complexity for your user; a new button on the dashboard, a new step in the process, a new setting or preference to adjust, more pages in the user manual to read. Think about it like this: Every new feature you implement adds weight to your product, weight that your users must carry with them while they

use your product; resistance. The right amount of extra weight makes your product and customer stronger and more powerful; but add too much extra resistance into the mix and you'll quickly exhaust your customers.

PRO TIP: View Every Feature as a Source of Resistance.

On the same note, too much extra weight can also exhaust your product if it's not engineered to carry it. If you get to a point where your product and customers are getting worn out and using your product for just the key function it was created for is no longer a simple task, it's probably time to clean out the excess features, they aren't adding value to your customer.

Forgotten Features

Outdated, Unused, & Insignificant Features

Of what use is a product with multiple, seldom-used, confusing and perhaps even obsolete features that consumers don't need or want? Hanging onto those features can (and will) cost your product and organization--why surrender users and/or revenue to a product feature that can just as easily be eliminated?

This one is simple: if a feature is not being used or making your company money, you've got a good reason to cut it. Not only will you save time and resources (such as support hours, engineering and QA hours between updates etc.) you'll also take some dead weight off your product.

Sunsetting a Feature

Before you cut any feature, you should first look at the implications of doing so; this means considering the impact the cut will have on your users as well as the technical implications.

A few preliminary considerations to make before cutting a feature:

- How many users are using the feature?
- Who uses it and how valuable are those users?
- Is it a paid feature? How much money is it making?
- What other areas of the product will be impacted?

TWEET THIS

Be prepared to manage backlash when cutting features

Much like launching a feature, phasing one out is a multi-team effort that will require coordination across your organization. At the very minimum, you'll need to work with your dev team who will not only be removing the feature but also ensuring that nothing else breaks in the process. In most cases you'll also need to coordinate with customer facing teams including marketing, support, and sales to ensure that everyone is prepared to communicate about the change as needed.

As is the case with any product change you make, you must be prepared to manage backlash from customers; this is especially important to remember when cutting features.

Angry customers won't hesitate to take to social media to complain, especially if they don't see any other outlet for their complaints, or if they feel as though they aren't being heard. Be sure your support team is not only prepared to field incoming complaints about a feature that's been cut, but is also prepared to track them.

PRO TIP: Be Prepared to Bring Features Back

In rare cases, you may have to bring a feature back--if enough customers complain about a feature you've cut, this may be a bridge you'll have to cross. No one likes being wrong, but as the cliche goes, "two wrongs don't make a right," and ignoring feedback from a substantial group of users is certainly not right.

This doesn't mean you have to bring back a feature just because customers complained; it means at the very least acknowledging the complaints and taking them into consideration; above all, empathy is the key.

Conclusion

Exceptional features are not haphazard, and rarely are great features designed, built, and launched overnight. The most successful features--the ones that customers use, love, and evangelize--are those that are built with customers in mind from the idea stage forward.

Product teams should not strive to crank out as many features as humanly possible; they need not get caught up in a features race, nor do they need to build Swiss Army Knives. Exceptional product teams should strive to exhibit laser sharp focus on product vision, company goals, and customer needs at all times throughout a feature's lifecycle; from the idea stage, to putting it on the product roadmap, to coming up with specs and building it, to testing and launching it, to (eventually) sunsetting it.

Use the Right Data to Build Features Customers Love

Let your customers help you know what to build next
with our leading product management platform

+ Leverage customer feedback to discover new opportunities and diagnose customer pain
+ Collect actionable data & easily prioritize feature requests
+ Engage your customers to validate your ideas

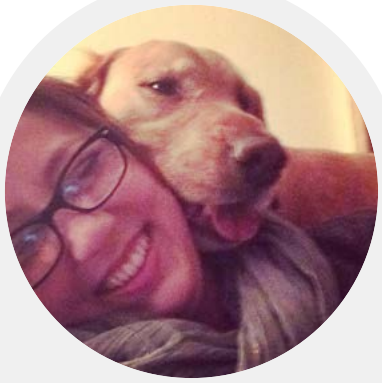
TRY IT!

Start your free
14-day trial today!

uservoice

Product Features: The Good, the Bad, & the Ugly

How to Prioritize, Build, Launch, and Maintain Features that Rock



BY HEATHER MCCLOSKEY

Inbound & Content Marketing Manager, UserVoice

Heather McCloskey is a former broadcast news producer turned product managment & marketing nerd. When she's not researching or writing, she can be found putting pedal to the metal behind a sewing machine or painting watercolor comics.

© 2015 UserVoice, Inc

uservoice

References

Thompson, D. V., Hamilton, R. W., & Rust, R. T. (2005). Feature Fatigue: When Product Capabilities Become Too Much of a Good Thing. Journal Of Marketing Research, 42(4), 431–442. http://doi.org/10.1509/jmkr.2005.42.4.431

Olchowka, Bartosz. Beware of features overload. (2014). Retrieved July 21, 2015, from https://developers.livechatinc.com/blog/beware-of-features-overload/

Crow, Dwight. Be Data-Informed, Not Data-Driven: Balancing Design, Qualitative Feedback, and Quantitative A/B Testing. (2015). Retrieved July 21, 2015, from https://community.uservoice.com/talks/be-data-informed-not-data-driven-balancing-design-qualitative-feedback-and-quantitative-ab-testing/

Saint-Exupéry Antoine de. (1939). Terre des hommes. Paris: Gallimard.

Helvetica - Gary Hustwit. Retrieved July 21, 2015, from http://www.hustwit.com/category/helvetica/

Bastow, Janna. To Kill a Feature - MindTheProduct. (2011). Retrieved July 21, 2015, from http://www.mindtheproduct.com/2011/08/to-kill-a-feature/